

OPS-SAT MBSE EXECUTIVE SUMMARY

Executive Summary

ESA Contract No. 4000134652/21/GLC/mk

APPROVAL

Title OPS-SAT MBSE Executive Summary	
Issue Number 1	Revision Number 0
Author Raphael FAUDOU	Date 04/06/2022
Approved By	Date of Approval

CHANGE LOG

Reason for change	Issue Nr.	Revision Number	Date
First version	1	0	06 April 2022

CHANGE RECORD

Issue Number	Revision Number		
Reason for change	Date	Pages	Paragraph(s)
1	0		

DISTRIBUTION

Name/Organisational Unit
ESA: Petros Pissias, Dominik Marszk, Tom Mladenov, Vladimir Zelenevskiy, Vasundhara Shiradhonkar, David Evans, Lorenzo Gagliardini, Marcus Wallum, Robin Biesbroek
AIRBUS DS: Marie-Hélène Deredempt, Jean-Luc Marty
Samares Engineering: Ida Dahl, Julie De Sousa, Raphael Faudou

Table of Contents

Executive Summary 1

1 Introduction 2

1.1 Document purpose..... 2

1.2 Specific terminology..... 2

1.3 Applicable and Reference Documents or Deliverables..... 3

1.4 Document organization..... 3

2 Activity objectives and work logic..... 3

2.1 Background..... 3

2.2 Activity objectives 3

2.3 Activity work logic 3

3 Tasks performed 4

3.1 Capture of pain points and selection of an MBSE solution able to address them..... 4

3.2 OPS-SAT reverse engineering (task 3)..... 6

3.3 Preparation of the OPS-SAT 2 mission (task 4) 7

Table of Illustrations

Figure 1: activity work logic 4

Figure 2: Airbus MBSE framework: MOFLT 5

Figure 3: MBSE Framework concepts in 5 layers 5

Figure 4: initial modeling approach for reverse engineering of OPS-SAT 6

Figure 5: granularity of functions identified from flight segment technical components..... 7

Figure 6: improvement of the modelling approach after end of task 3..... 8

Figure 7: simulation of the operational concept for OPS-SAT 2..... 8

Figure 74: OPS-SAT 2 simulation of several experiments - test, upload, installation and execution on the S/C..... 9

Figure 8: illustration of the ability to preview different baselines in the same model..... 10

Figure 9: recommended approach for OPS-SAT 2 design..... 10

1 INTRODUCTION

1.1 Document purpose

The purpose of this document is to give a short summary of the activity that has been performed and the achievements that were obtained from it.

1.2 Specific terminology

R-MOFLT	Requirements, Mission, Operations, Functions, Logical, Technical
CSM	Cameo Systems Modeler software tool
Plugin	Additional software on top of the CSM tool

--	--

1.3 Applicable and Reference Documents or Deliverables

TN-01	Technical Note about the capture of pain points and lesson learned
SU-URM	User requirements of the MBSE Framework

1.4 Document organization

The chapter 2 recalls the background and activity objectives
 The chapter 3 presents the suggested approach and work logic
 The chapter 4 details the tasks performed during the activity
 The chapter 5 presents a conclusion with achievements and lessons learned.

2 ACTIVITY OBJECTIVES AND WORK LOGIC

2.1 Background

After the launch of the OPS-SAT mission on 18 December 2019, it was possible to get first feedback from the team. This feedback has revealed some pain points and challenges during the different stages before the launch, especially for the design, assembly, verification, validation, and even operations. Before future envisioned IOD missions through new “OPS-SAT” programs (including OPS-SAT 2 mission from 2024), there is a slot that can be used to try to address the pain points and challenges of the OPS-SAT team.
 A system definition model, supported by both an MBSE methodology and a modelling tool, is seen as a potential enabler to address part of those pain points and challenges.

2.2 Activity objectives

The activity had the 3 main objectives:

1. Identify the MBSE added value compared to the traditional document centric approach while avoiding all mistakes captured in previous MBSE deployments in the space sector (and others)
2. Select an MBSE tool and a methodology to use it in operational projects with an agile, both top-down and bottom-up design, and considering the pain points and challenges identified by OPS-SAT engineers.
3. Produce an OPS-SAT model from OPS-SAT document pack, check that this model can improve some of the pain points identified by OPS-SAT team and derive a system reference model usable for future IOD missions including OPS-SAT 2 mission.

2.3 Activity work logic

The work logic proposed by ESA was the following:

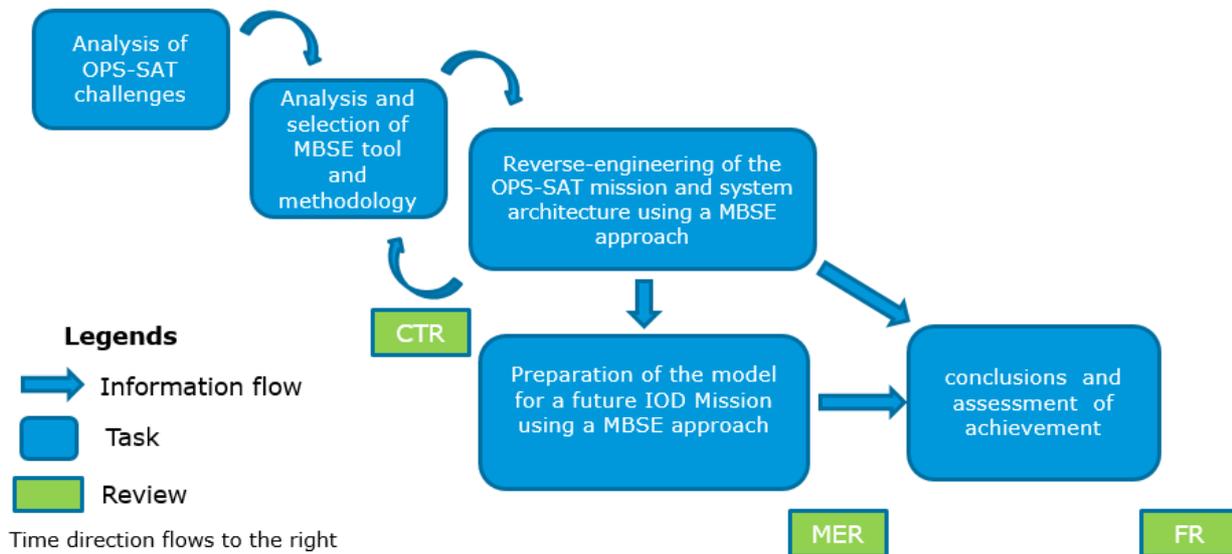


Figure 1: activity work logic

3 TASKS PERFORMED

3.1 Capture of pain points and selection of an MBSE solution able to address them

After a capture of the pain points and lessons learned from OPS-SAT team through several workshops with ESA and TU Graz (prime of OPS-SAT system), we have identified a set of modelling tracks to improve those pain points and challenges.

With the support of Airbus DS (contractor), we have then presented an MBSE framework developed by Airbus and their partners on top of the Cameo Systems Modeler tool (SysML tool developed by NoMagic company and acquired 2 years ago by Dassault Systeme). This MBSE tool was selected by Airbus a few years ago with a formal selection process¹.

This framework hides the SysML notation (often found complex by systems engineers) with an extension called SECAM (Systems Engineering Common Architecture Model) that was initially built by Airbus DS to address ECSS standards (vocabulary and processes). A large part of this SECAM model (or meta model) has been proposed by Airbus DS to the ESA OSMOSE project to align all space domain entities on a common Systems Engineering vocabulary.

The Airbus MBSE framework, also called “MOFLT” for Mission, Operations, Functional, Logical and Technical, provides SECAM concepts (implemented on top of SysML notation) and a set of methods to develop any system (including spacecraft) and conformant to the ISO 15288 standard processes for systems engineering as illustrated below:

¹¹ Details on this MBSE tool selection process are available in TN02.

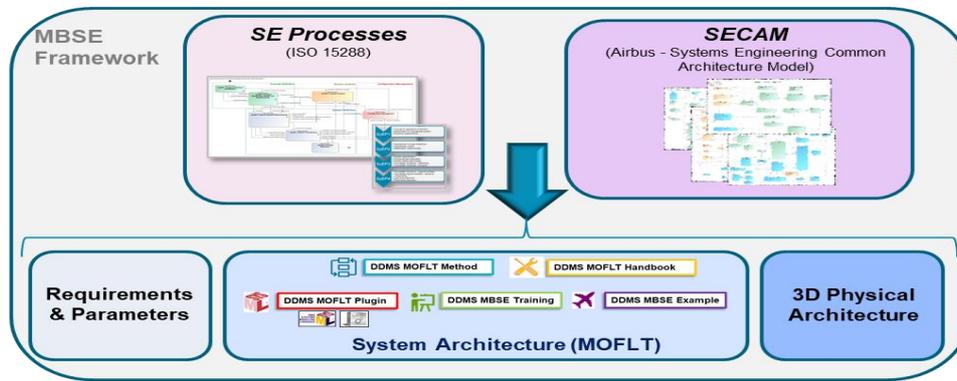


Figure 2: Airbus MBSE framework: MOFLT

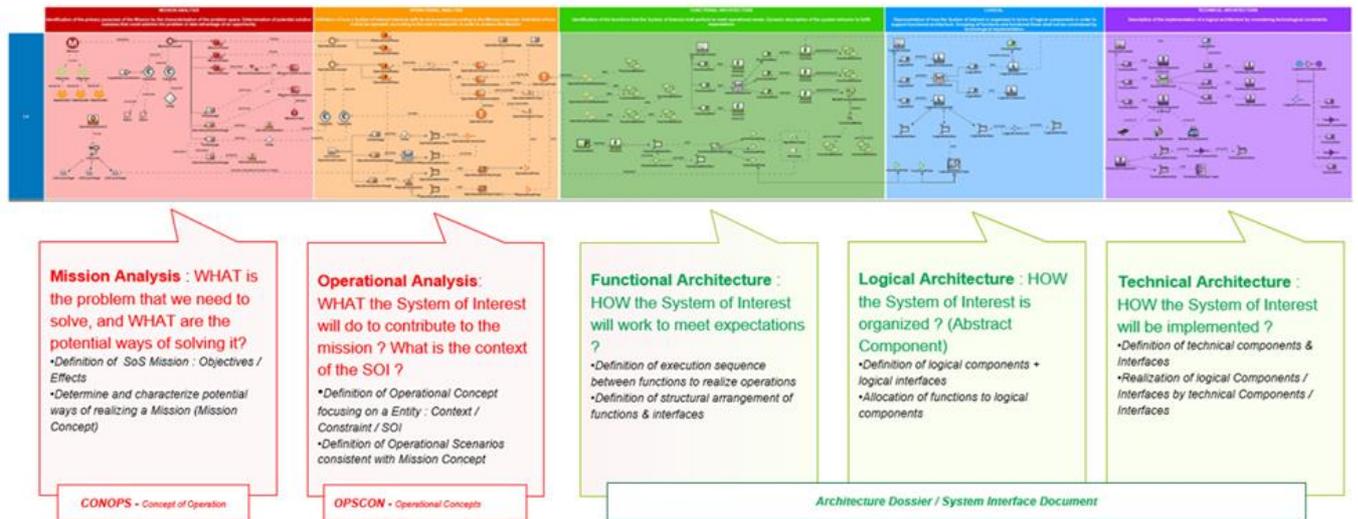
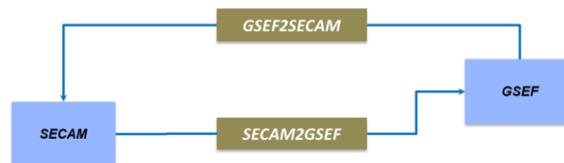


Figure 3: MBSE Framework concepts in 5 layers

We have also analyzed ESA GSEF² initiative and have compared it to SECAM. We have presented this comparison to ESA during a workshop and we have concluded that they share many similarities but GSEF is more focused on the technical architecture while SECAM has a larger scope with mission, operations, functions and logical layers not defined in GSEF. Both SECAM and GSEF have mapping concepts to ensure interoperability with each other:



We have analysed all the modelling needs derived from the pain points and challenges shared by the OPS-SAT team and showed that the selected MBSE Framework was able to address all of them except 2:

- MBSEF-ON09: Need for modeling of behavior of a transponder.
- MBSEF-ON10: Modeling of dysfunctional behavior and analysis of errors and their propagation → dysfunctional behavior can be formalized with the framework but there is

² Ground Segment Engineering Framework

not support currently in the analysis of errors and their propagation using simulation. It would require some additional automation plugins to analyze the simulation traces.

Note: All details of the identification of MBSE framework requirements is available in **SU-URM**.

3.2 OPS-SAT reverse engineering (task 3)

The initial reverse engineering approach consisted in translating the different information collected in the various documents to initialize the model with 2 levels of systems:

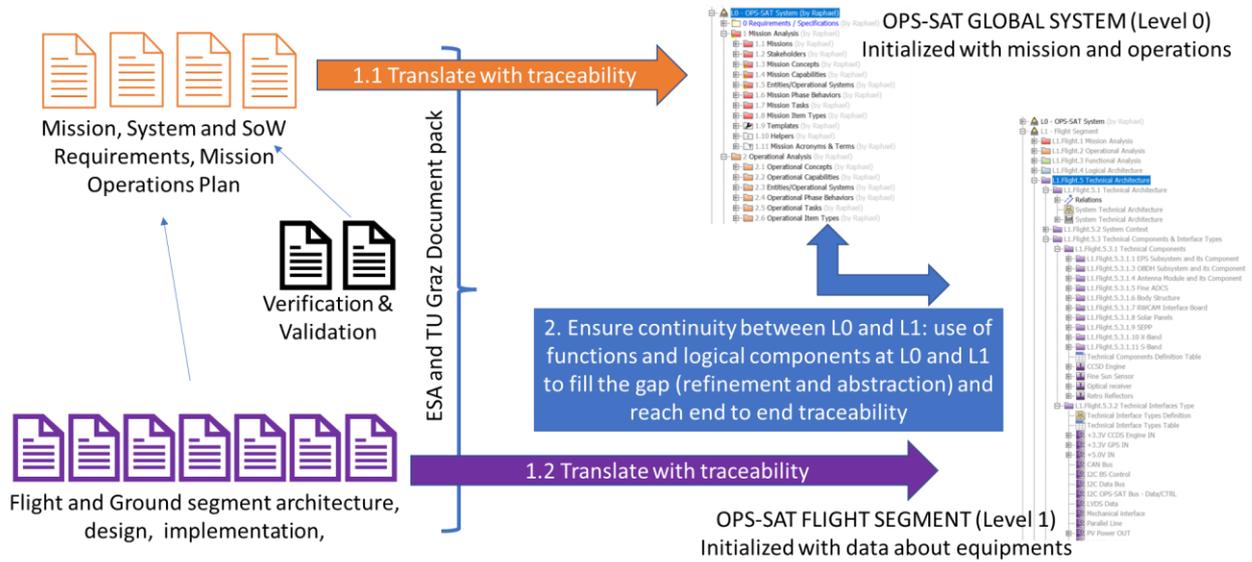


Figure 4: initial modeling approach for reverse engineering of OPS-SAT

After the 2 iterations we were able to demonstrate that the MBSE framework selected could support most of the modelling needs derived from the pain points of the OPS-SAT team:

- end to end traceability between the mission and system reqts down to technical components,
- ability to detect integration issues with incompatible connectors,
- ability to build a unique technical architecture with several views on electrical, mechanical, buses integration, 2D geometry
- ability to generate documentation,
- ability to provide an easy navigation between key diagrams
- ...

The major difficulty was to find the right level of granularity for logical elements like **functions and logical components**, able to fill the gap between global system intended operations (top) and the description of the physical equipments and their integration (bottom).

The first attempts to define logical components and functions by abstraction of physical equipments consumed a lot of efforts to gather all the information (with need of extra documents under NDA) and to decide what to remove during the abstraction transformation. We dived too deep in the details and got technical functions, too far from the mission and operations to ensure a possible digital continuity, as illustrated below.

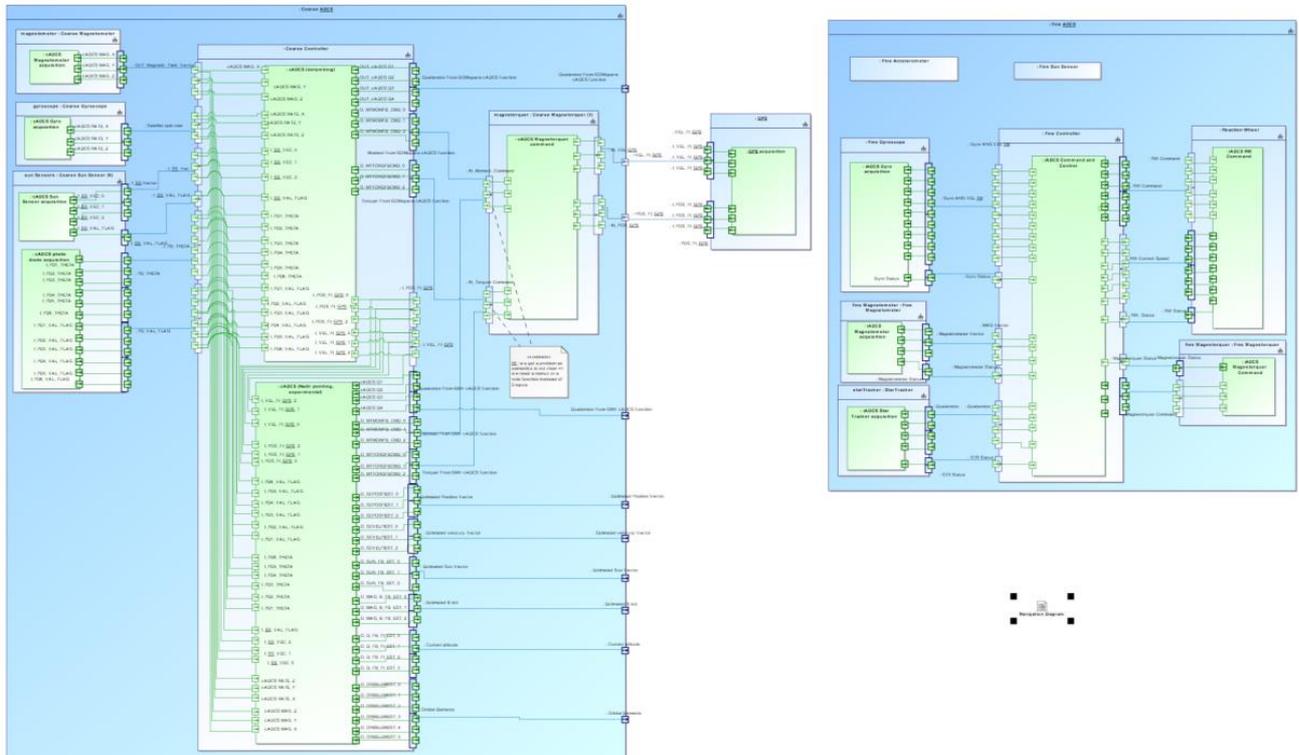


Figure 5: granularity of functions identified from flight segment technical components

After the end of the reverse engineering (task T3) we have had confirmation that it was hard to find the intermediate engineering artefact (functions and logical components) able to fill the gap between Level 0 operations and Level 1 technical components. We were able to derive some system functions from the description of operational procedures (top-down approach) but **it was extremely hard to abstract the equipment functions into system-level functions (bottom-up).**

We have also highlighted that the efforts to build a same technical architecture with several views like “electrical”, “mechanical” and “buses integration” were quite high when built from scratch with **documents detailing the physical products datasheets. Indeed** it requires a lot of specific knowledge (specialty engineering) to understand what is useful or not in a big set of data and to check the consistency between all the various documents with sometimes different names for the same equipment according to its role...

3.3 Preparation of the OPS-SAT 2 mission (task 4)

For the start of the task 4 (preparation of future missions), we have decided to **introduce a set of functions translated from the space domain knowledge (with Airbus support) to ease the continuity between L0 operations and L1 technical components (flight and ground).**

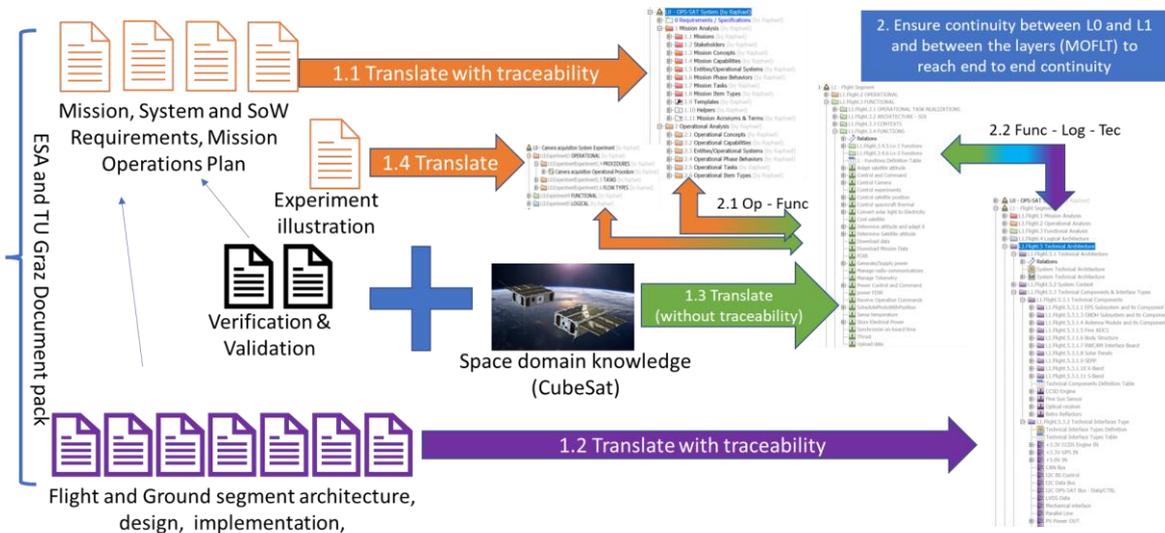


Figure 6: improvement of the modelling approach after end of task 3

We have also benefited from a new document (OPS-SAT 2 study report) with a lot of information about the OPS-SAT 2 mission and its intended operations. It gave us the ability to refine the operational behavior of the Spacecraft, of the Mission Control System, and some of the experimenter interface. We were then able to build the OPS-SAT 2 integrated model, made of the assembly of the level 1 systems and showing the operational exchanges between them over time.

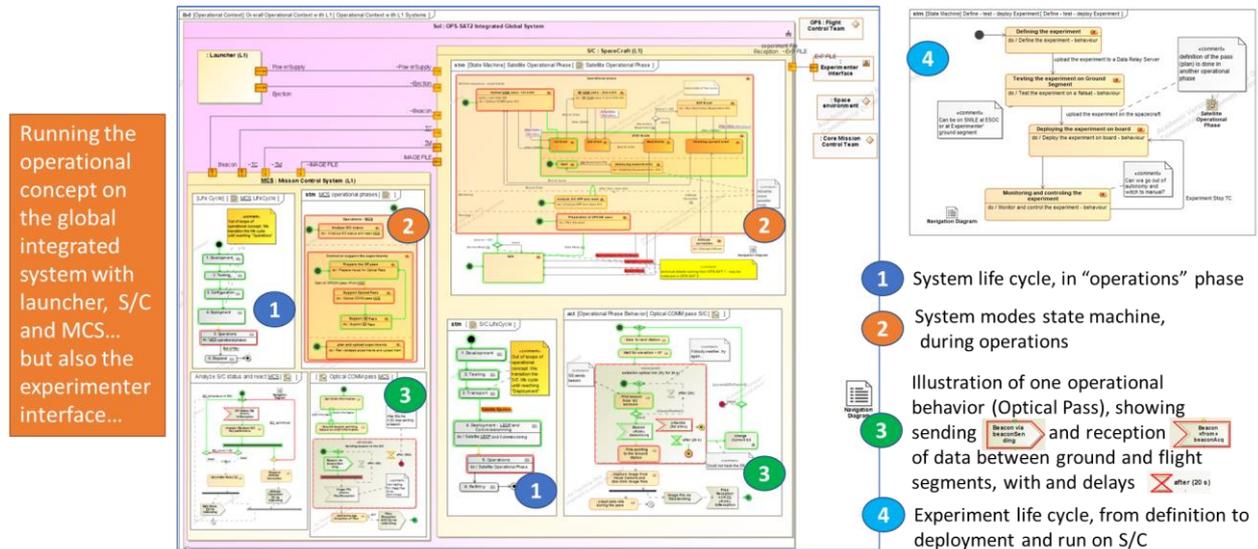


Figure 7: simulation of the operational concept for OPS-SAT 2

The simulation of the global operational architecture (integrated system model) was extremely useful to detect issues in some operational scenarios (missing acknowledge, missing timeout, unclear time for synchronization, case leading to an infinite loop...) and check the behavior. **We consider this executable operational model as a toolbox to support the early validation of the operational concept.**

We were able to show the simulation of several different experiments (as displayed below in blue and orange in the console). The simulation showed the deployment of the new experiment, starting from the experimenter interface, then updated and installed on the spacecraft, and finally run during the next "experiment" pass.

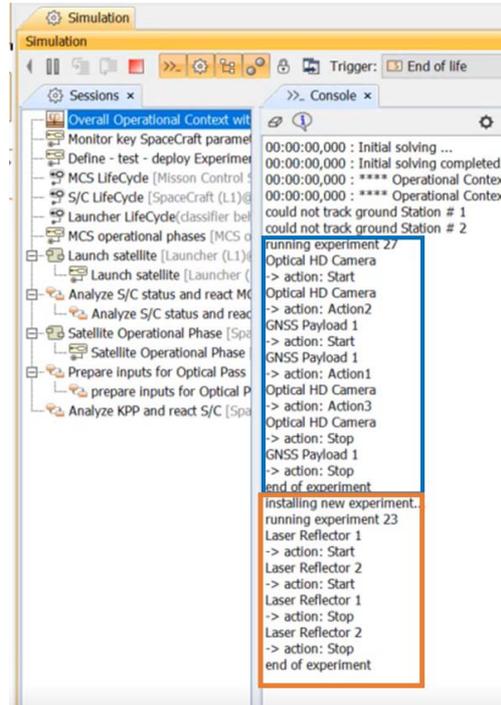


Figure 8: OPS-SAT 2 simulation of several experiments - test, upload, installation and execution on the S/C

The new document about OPS-SAT 2 mission mentioned 2 configurations (6U and 12U). With 6U there is no place for propulsion (thrusters) or spring-deployed drag sails.

We have shown that it was possible to create a 150% model (a model that combines several options) with variable elements displayed in green when available in the given selected configuration and in red when not selected according to the chosen configuration.

#	Name	Star Tracker Alignment : Alignment Type	Propulsion : Boolean	Deployable passive drag : Boolean	Platform capacity : Platform Capacity	Real time orbit determination : Boolean
1	6U design	Along Cardinal Directions	<input type="checkbox"/> false	<input type="checkbox"/> false	Low	<undefined>
2	12U design	Along Non-Cardinal Directions	<input checked="" type="checkbox"/> true	<input checked="" type="checkbox"/> true	High	<undefined>

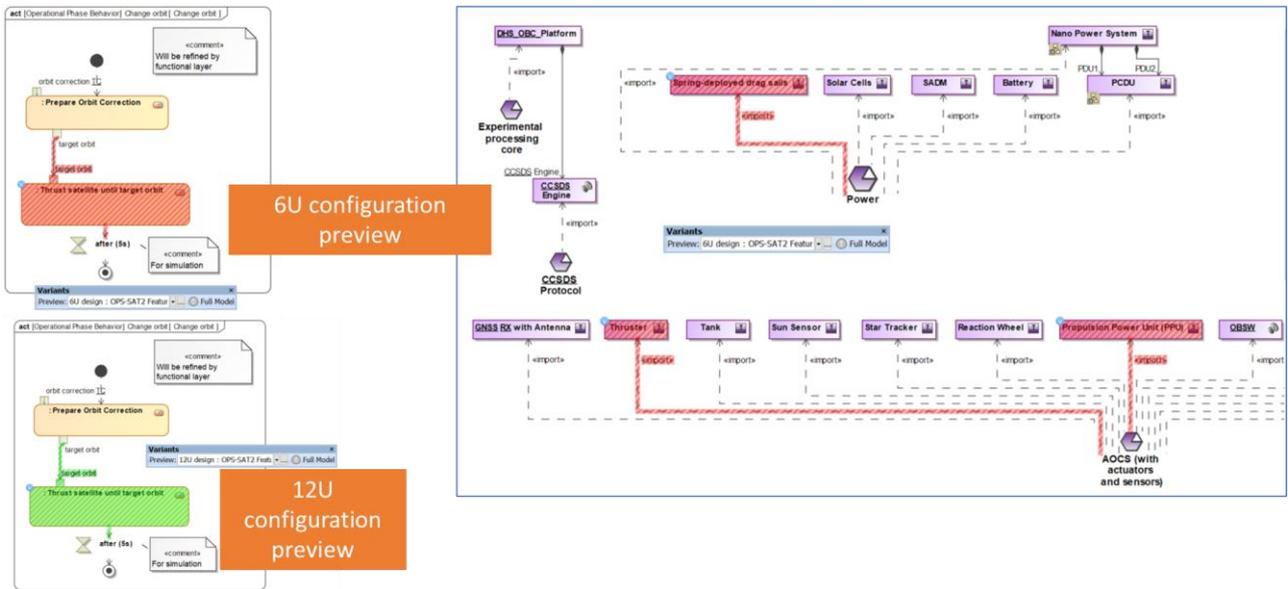


Figure 9: illustration of the ability to preview different baselines in the same model

The availability of flight segment functions as input functions allowed to break the complexity of the initial gap (between OPS-SAT operations and Flight segment technical components) into several smaller gaps with clear transformation semantics:

1. Refinement and allocation of OPS-SAT (Level L0) operations into Flight Segment (L1) operations and Ground System (L1) operations and simulation
2. Allocation of Flight Segment functions to a first set of Flight Segment logical components
3. Adaptation of flight segment logical components to ensure coverage (abstraction) of the flight segment technical components
4. Adaptation of Flight segment functions to support Flight segment operations (and same for Ground Segment)

By adding feedback from the integrated model and its operational architecture simulation it gives us the following recommended approach for OPS-SAT 2 modeling:

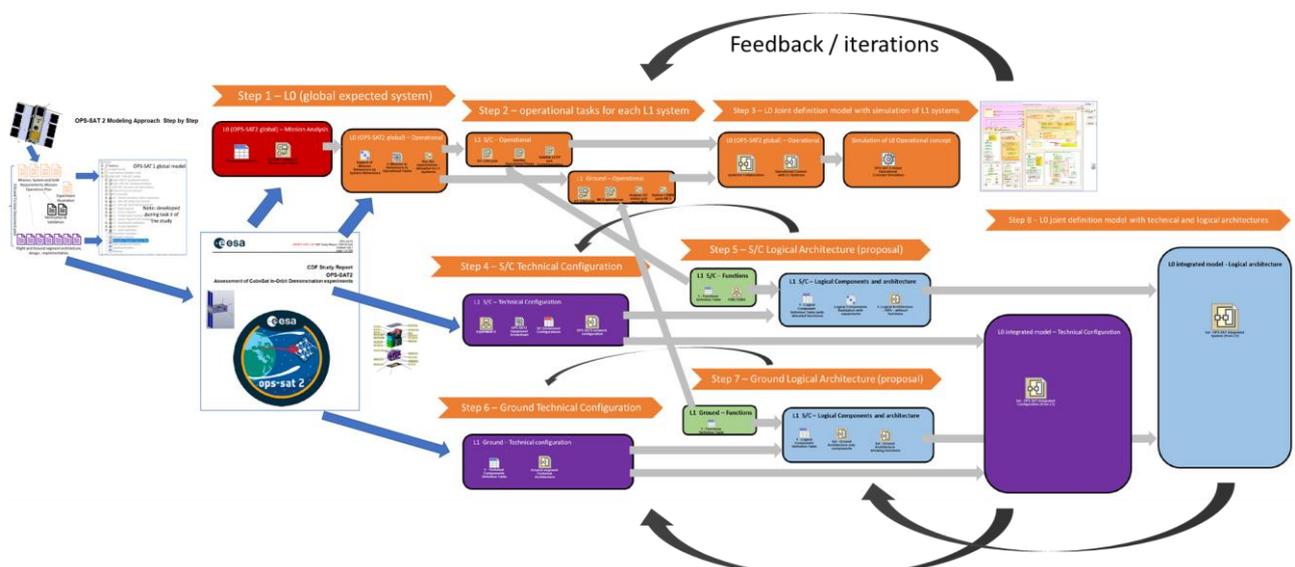


Figure 10: recommended approach for OPS-SAT 2 design